Collaborative Filtering Using a Regression-Based Approach and Classification-Based Approach

Roll Number: 1401119
Akshat Doshi
School Of Engineering & Applied Sciences, Ahmedabad University.

***Abstract*—The world is connect through the power of social media platform. People want know what other people are doing and according to that, they adapt themselves to challenge the new world. People want to acquire new skills according to their job carrier path. Skill seeking has been a tricky, tedious and time consuming task, because people looking for a new opportunity had to collect information from many different sources this type of system is required. In this report skill, recommendation systems according to carrier path has proposed in order to automate and simplify this task, also increasing its effectiveness. However, current approaches rely on scarce manually collected data that often do not completely reveal people skills. Our work aims to find out relationships between jobs and people skills.**

***Keywords*– Deep Neural Network, Collaborative Filtering, Linear regression , Mean Normalization, Gradient Decent, Regularization.**

I. Collaborative Filtering

The recommendation systems are classified according to the technique used to create a recommendation: Content-based system, Collaborative-based system and Hybrid-based system. In this report we have used Collaborative-based recommendation systems.

Collaborative filtering is one of the most successful recommendation technologies. The basic idea behind this wonderful technology is to build a community of users and recommend items to a certain user according to many similar users preferences. Collaborative filtering have two fundamental steps in prediction: first, some users, known as neighbors, are selected due to their similarities to the active user; second, a weighted average of neighbors ratings are used to predict the rating value. Based on these relationships, recommender systems can make more accurate predictions without compromise of system performance, which means not only lower Mean Absolute Error (MAE) but also good user experience.

The rest of the paper is organized as the follows. The next section would desicire our dataset. The section 3 describe newly proposed approach based on linearregression. In section 4 we propose another approach base on Restricted Boltzmann Machines (RBM) using neural networks. In section 5 we introduce the ways of evaluation and conclusion.

II. Data set

We have 39 different job position is given in Json format. In this there is different type of information of many user such as candidateID , skill, work position, experience etc. So for cleaning the data we firstly convert Json file into CSV format and that remove comma, dot and another unrelated information and save new CSV file with information that we required. Now the data is like this in first column we have candidateID, in second column we have skills of all the candidate and in last column we have experience of the user in year. The first difficulty that we have faced is to clean the data for our requirement.

This type of system can me modeled as Content-based system, Collaborative-based system, Hybrid-based system, Latent semantic analysis (LSA), fuzzy logic, string matching, clustering, classification, using probabilistic models and many more. So we restrict our boundary and model this approach into Collaborative based system and classification based system. So now we will both the methods in detail.

III. Approach 1

In this approach we have to predict the skills of candidate using neighbourhood users skills data. A representation for collaborative filtering tasks that allows the application of virtually any machine-learning algorithm.

To find the skill of any particular candidate I have use collaborative learning algorithm. The first step is to collect the skills of the users. Our Collaborative Filtering (CF) implementation stores the data in two 2D matrices. So for each skill we took ones or zeros (have that skill or not have that skill) in a row and each user is in a column. This type matrix is quite sparse, since not all users have all the skills.

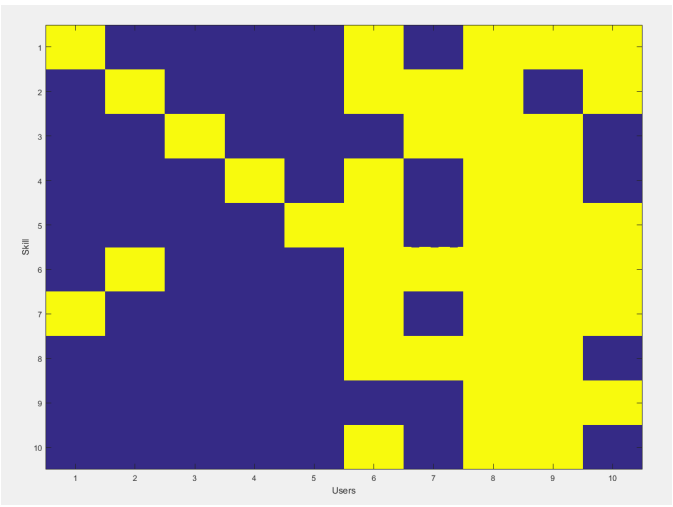

Fig 1: Sparse matrix (black boxes is 1's and white boxes is 0's

After getting the 2D dataset matrix (skills X users), we implement Collaborative Filtering based recommendation system model. For the model, we need to compute the top recommended skills. In the following part, we will explicitly show how we build our recommendation system, and compare different models in the following subsection.

A. Module-1

In our first module code, we assume that we have given job title and we have 10 users and total 10 skills are required for that particular job. So initial our data matrix Y (as seen in fig 1) exists in the form of a sparse matrix, where rows correspond to skills, columns correspond to user and the matrix entries are either zeros or ones. Then we have feature vector of experience. In that we map X matrix as average experience required of that particular job and Theta vector as experience of each user. Now we have predict the skill weather it is required or not for a particular user in that given matrix. So in order to solve a problem we have used linear regression method. We got X matrix (Skill feature) and Theta matrix (user's feature) by using this matrices we perform linear regression. As, linear regression focuses on to minimize this cost function and minimize the sum of squared errors. After multiple iterations of gradient descent, we would have found the user feature and experience feature matrices that minimize our cost function. Essentially, we have learned the appropriate values of user and experience to make accurate predictions on the skills for every user.

Now, lets use our learning algorithm to predict top skills for each user. We just have enter the candidate ID and our predictor will predict the skills.

```
Recommender system learning completed.

Top skill recommendations for user ID 7 :
Predicting top new skill for future : Hadoop
Predicting top new skill for future : Db2
Predicting top new skill for future : C
Predicting top new skill for future : Python
Predicting top new skill for future : Oracle
Predicting top new skill for future : C++
Elapsed time is 0.757283 seconds.
```

Fig 2: Module 1 output for candidate ID - 7

B. Module-2

Second module is the online version of the first module where new user will come who have or not have any required skill preference for that particular job. So, for that he/she wants to acquire skill for that particular job position. This module also have same step as above module. Firstly, we have 10-by-10 sparse matrix which column contains each users and row has each skills which is needed. Now new user has come now we will assign the required parameters like skills that particular new user have, new user feature and skill feature. Then, we calculate linear regression followed by gradient decent to minimize our cost function and minimize the sum of squared errors.

```
Recommender system learning completed.

Top skill recommendations for you:
Predicting top new skill for future : Hadoop
Predicting top new skill for future : Java
Predicting top new skill for future : Db2
Predicting top new skill for future : C
Predicting top new skill for future : ESD
Predicting top new skill for future : Python
Predicting top new skill for future : Sql
Predicting top new skill for future : Oracle
Predicting top new skill for future : C++
Elapsed time is 0.466123 seconds.
```

Fig 3: Module 2 output for candidate ID - 1 (new user)

Now, lets use our learning algorithm learn to predict top skills for this new job position. We just have enter the new candidate ID and our predictor will predict the skills based on our learning algorithm.

C. Algorithm for approach 1

Algorithm 1 Collaborating Filtering using Linear Regression

1: for every skill i that user u has some or no preference
2: for every other user y that has a preference for i
3: compute a similarity s between u and y
4: add y's preference for i, weighted by s, to a running average
5: return the top skills, ranked by weighted average

IV. Approach 2

As we can see from above approach, it is basically a classification problem so linear regression will fail for classification problem. So for classification problem we have many approach like logistic regression, K nearest neighborhood, K-means, K-medians, Support vector machine, neural network etc.

In this approach-2 I have used Restricted Boltzmann Machines (RBM) essentially perform a binary version of factor analysis. This is one way of thinking about RBMs although there are many different ways to use RBMs. Instead of users features we will map the skills on a continuous scale, they simply tell you whether they have this particular skill or not by ones and zeros, and the RBM will try to discover latent factors that can explain the activation of these skill choices. A Restricted Boltzmann Machine is a stochastic neural network. In neural network, we have neuron-like units whose binary activation function weight depend on their connected neighbors. The stochastic neural network consist of,

1) First layer of visible units (users skill preferences whose states we know and set);
2) layers of hidden units (the latent factors we try to learn)
3) A bias unit (whose state is always on, and is a way of adjusting for the different inherent popularity of each skill).

Restricted Boltzmann Machines and neural networks in general, works by updating the weights of the neurons in

multilayer feed forward neural networks who uses logistic activation function and back propagation to minimize the error.

If one job position required six skills and we have six users then our data matrix will become 6-by-6 data matrix whose row contains user and column contains skills. Therefore, the six skills will send messages to the hidden units, telling them to update weights themselves and by using backpropagation method we will iteratively change the weights of the neuron until the network converges (i.e., the error between the training examples and their reconstructions falls below some threshold) or we reach some maximum number of epochs.

```
Weights
[[  7.98605702e-01   1.09638691e+00  -8.68297237e-02  -4.52542722e-03
   -2.90567644e-01   1.35371887e+00   1.47951119e+00]
 [  7.82168630e-01  -8.41655644e-01   4.19671989e+00  -3.41679768e+00
   -4.05051153e+00  -1.56546239e+00   3.71834866e+00]
 [  1.16455545e+00  -6.92660775e+00   2.61620547e+00  -3.21515579e+00
   -3.15490719e+00  -6.33216635e+00   2.32488367e+00]
 [  2.55200205e+00   2.49270740e+00   1.41151422e+00   1.19454430e+00
    1.12876436e+00   2.64937475e+00   3.23609816e+00]
 [ -8.26736769e-01   1.26290459e+00  -5.22629212e+00   2.75004838e+00
    3.24328274e+00   1.89445206e+00  -2.80361419e+00]
 [ -6.96814663e+00   1.07844462e+00  -3.62871255e+00   1.18448511e+00
    1.75210648e+00   1.56294163e+00  -5.92468893e+00]
 [ -2.24780701e+00  -1.77325318e+00  -1.98257944e+00  -1.64279011e+00
   -1.59279417e+00  -1.57416622e+00  -2.07264048e+00]]
Output as skill
[[ 1.  0.  1.  1.  1.  0.]]
```

Fig 4: Output as skill recommender

We have take six skills like java, C, C++, Matlab, Python and R for recommendation. We have given training data as sparse matrix containing ones and zeros that denotes that user have or not have that skill. So by our RBM neural network algorithm we got the weights of our multilayer neural network and according to that we predict the skill for the new user we have or don't have any skills.

V. Evaluation and Conclusion

In approach 1 statistical accuracy metrics measure how close the predicted ratings (experience) generated by various kinds of algorithms are to actual user ratings (experience). Mean Absolute Error the average absolute error between the predicted rating and the actual skill experience given by a user and from that we predict the skills of the other or a new user. This method is one of the most widely used predictive accuracy metric in evaluation of recommender systems. The major drawback of this system is, actually this skill recomandation system is based on classification problem and linear regression is only use for linear relationship between a dependent variable and one or more independent variables. So to map our algorithm in this fashion have less change to get perfect output but as we can use linear regression ass classification problem by using threshold parameter.

As we can see from below image our online linear regression give this type of output graph so from output graph we can see that most of our predictor line is between 0.4 to 0.5. So by setting threshold parameter between 0.4 to 0.5 we could predict good outcome.

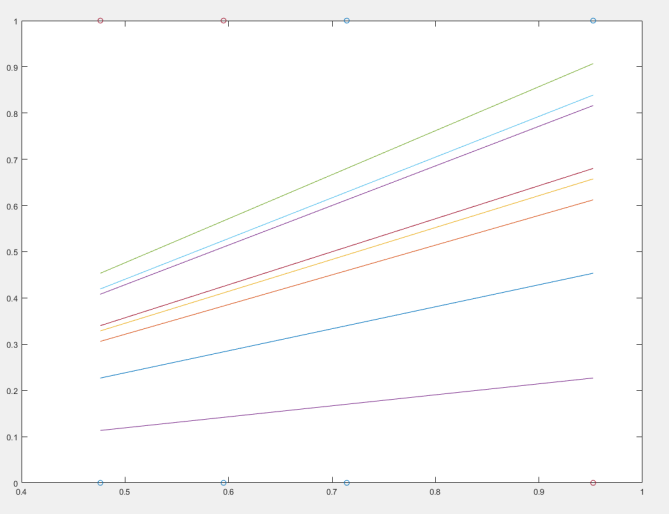

Fig 5: Approach 1 linear regression online version output

As failure of approach 1 we shift our algorithm to approach 2. We see that our problem is actually classification problem so in order to solve that we have use neural network approach. It will good output accuracy then approach 1 but as dimension increases the time taken by algorithm will increase. So in time complexity approach 1 is batter then approach 2 but for accuracy approach 2 is more batter then approach 1.

References

[1] A New Prediction Approach Based on Linear Regression for Collaborative Filtering, Xinyang Ge, Jia Liu*, Qi Qi, Zhenyu Chen State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China.

[2] Learning Collaborative Information Filters, Daniel Billsus and Michael J. Pazzani, Department of Information and Computer Science University of California, Irvine.

[3] Job recommendation system from semantic similarity of linkedin users' skill, by Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, Roberto Pasolini.

[4] The Browsemaps: Collaborative Filtering at LinkedIn, by Lili Wu, Sam Shah, Sean Choi, Mitul Tiwari, Christian Posse.

[5] https://nikhilwins.wordpress.com/2015/09/18/movie-recommendations-how-does-netflix-do-it-a-9-step-coding-intuitive-guide-into-collaborative-filtering/

[6] https://www.coursera.org/learn/machine-learning/lecture/2WoBV/collaborative-filtering

[7] https://github.com/echen/restricted-boltzmann-machines